Word Co-Occurrence in Hadoop

Write a MapReduce program in Hadoop that computes word co-occurrences. The key idea is to count the number of times each pair of words w_i and w_j occurs in a collection of text files.

PRELIMINARIES – You were provided with a VM for the labs and for the final project. If you have not done it yet, install Hadoop on that VM. The installation mode must be the *pseudo-distributed mode* (a.k.a *single-node cluster*). To this purpose, consider the guide available at: https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html. You can also find useful the tutorial and related recording that I explained in "*Lesson 4 – Hadoop Installation*".

THE DATA -- For this project, download and unzip the snippets.zip input archive. The snippets folder contains 782 text files with name `lineXXX`, where `XXX` runs from 000 to 781. Each input file contains a text on multiple lines, including numbers, punctuation, etc. Split every line (i.e., a record for the `map()` function) in every text file into "tokens" (i.e., space-delimited sequences of characters).

YOUR TASK – In this project you must:

- Consider the pseudo-code for the **Pairs** design pattern, which I presented in "*Lesson 5 – Design patterns*". Given a word w_i, its neighbor w_j is the term that immediately follows w_i (i.e., the next token);
- Implement the above described MapReduce algorithm using the Hadoop framework;
- Test your implementation in the cluster;
- Write a short project report detailing the implementation and experimental results.

FOR HIGHER MARKS – Examples to improve your project include, but are not limited to:

- Consider the **Stripes** design pattern (instead of the Pairs one) and a wider definition of a neighbor of a term (e.g., you can consider a window of terms that precede and follow w_i; or you can consider all the terms in the same sentence of w_i);
- Use combiners and/or more than one reducer;
- Use the `Mapper` and `Reducer` classes `setup()` and/or `cleanup()` methods.